

C++ ASSIGNMENT

1.Ques :Calculate the sum of the elements of nums between indices left and right inclusive where $\text{left} \leq \text{right}$.

Ans:

```
class NumArray {
public:
    vector<int> pre;
    NumArray(vector<int>& nums) {
        int n=nums.size();
        pre=vector<int> (n);
        pre[0]=nums[0];
        for(int i=1;i<n;i++){
            pre[i]=pre[i-1]+nums[i];
        }
    }
    int sumRange(int left, int right) {
        if(left==0) return pre[right];
        return pre[right]-pre[left-1];
    }
};
```

2.Ques :Given an array of integers nums , calculate the pivot index of this array. The pivot index is the index where the sum of all the numbers strictly to the left of the index is equal to the sum of all the numbers strictly to the index's right. If the index is on the left edge of the array, then the left sum is 0 because there are no elements

to the left. This also applies to the right edge of the array.

Return the leftmost pivot index. If no such index exists, return -1 .

[Leetcode 724]

Example 1:

Input: nums = [1,7,3,6,5,6]

Output: 3

```

Ans: class Solution {
public:
    int pivotIndex(vector<int>& nums) {
        int n=nums.size();
        for(int i=0;i<n;i++){
            int rightsum+=nums[i];
        }
        int leftsum=0;
        for(int i=0;i<n;i++){
            rightsum-=nums[i];
            if(rightsum==leftsum) return i;
            leftsum+=nums[i];
        }
        return -1;
    }
};

```

3.Ques :We define the conversion array conver of an array arr as follows:

$\text{conver}[i] = \text{arr}[i] + \max(\text{arr}[0..i])$ where $\max(\text{arr}[0..i])$ is the maximum value of $\text{arr}[j]$ over $0 \leq j \leq i$.

We also define the score of an array arr as the sum of the values of the conversion array of arr .Given a 0-indexed integer array nums of length n , return an array ans of length n where $\text{ans}[i]$ is the score of the prefix $\text{nums}[0..i]$. [Leetcode 2640]

Example 1:

Input: nums = [2,3,7,5,10]

Output: [4,10,24,36,56]

```

Ans: class Solution {
public:
    vector<long long> findPrefixScore(vector<int>& nums) {

```

```

int n=nums.size();
vector<long long> ans;
long long maximum= 0LL,s=0LL;
for(int x: nums){
    maximum= max(maximum,(long long) x);
    s+=(x+maximum);
    ans.push_back(s);
}
return ans;
}
};

```

4.Ques:There are n flights that are labeled from 1 to n .

You are given an array of flight bookings bookings , where bookings[i] = [first i, last i,seats i] represents a booking for flights firsti through lasti (inclusive) with seats i seats reserved for each flight in the range.

Return an array answer of length n , where answer[i] is the total number of seats reserved for flight i . [Leetcode 1109]

Example 1:

Input: bookings = [[1,2,10],[2,3,20],[2,5,25]], n = 5

Output: [10,55,45,25,25]

Ans: class Solution {

public:

```

vector<int> corpFlightBookings(vector<vector<int>>& bookings,
int n) {
    vector<int> res(n,0);
    for(int i=0;i<bookings.size();i++){
        res[bookings[i][0]-1]+=bookings[i][2];
        if(bookings[i][1]<n) res[bookings[i][1]]-=bookings[i][2];
    }
}

```

```
    for(int i=1;i<n;i++){  
        res[i]+=res[i-1];  
    }  
    return res;  
}  
};
```